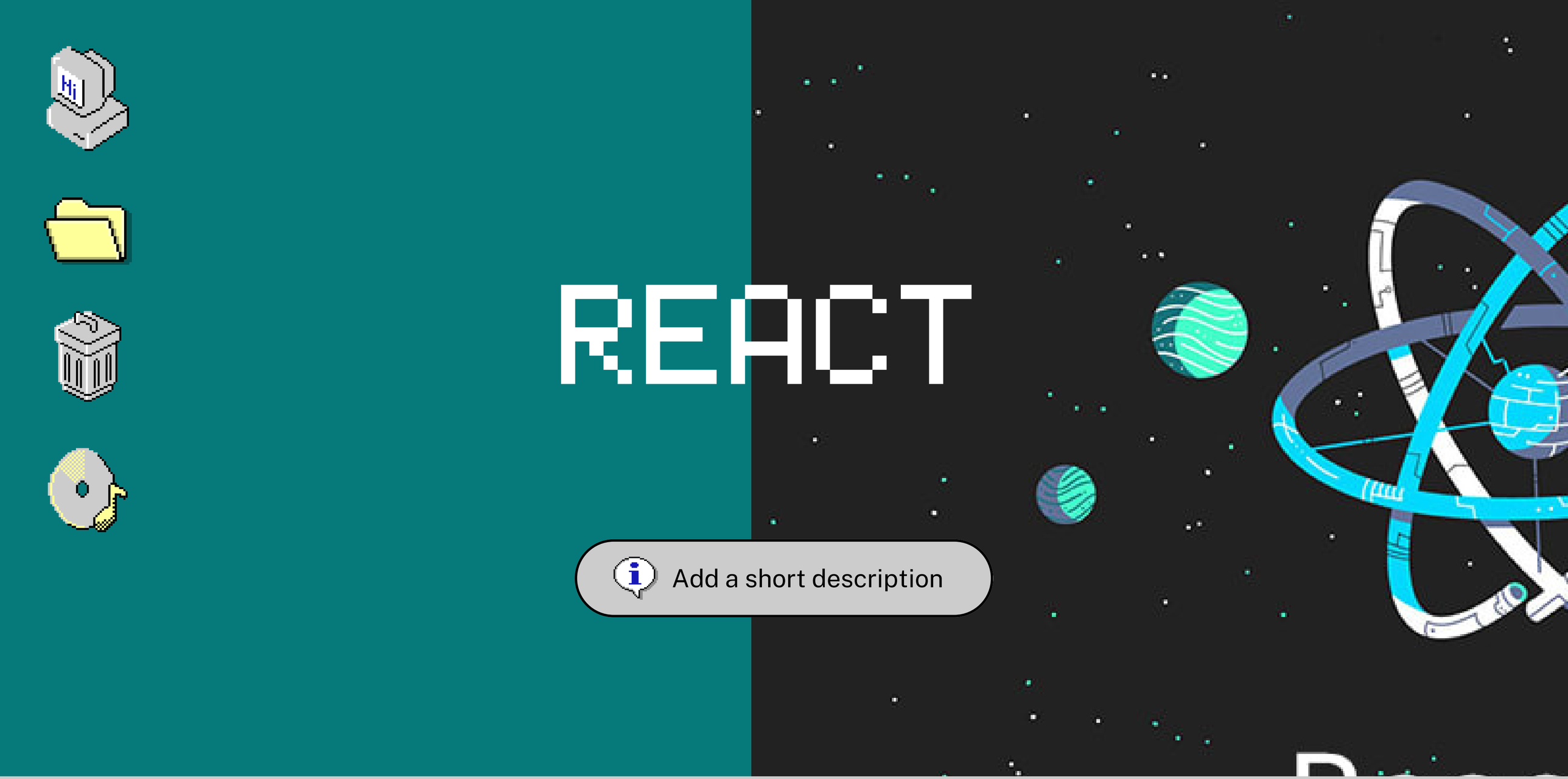
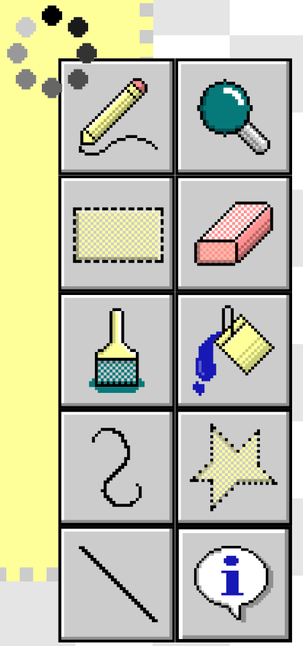
REACT



Add a short description

11:11PM

INSTALACION



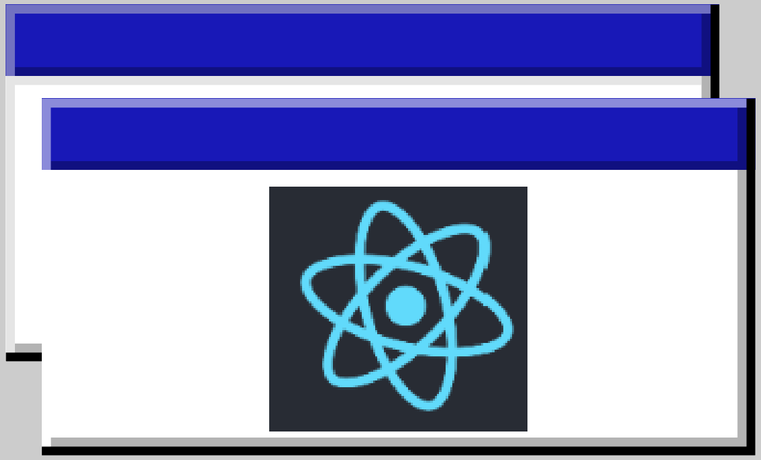
Back to Agenda Page

Instalar Node Js

node -v

Instalar npm

npm -v

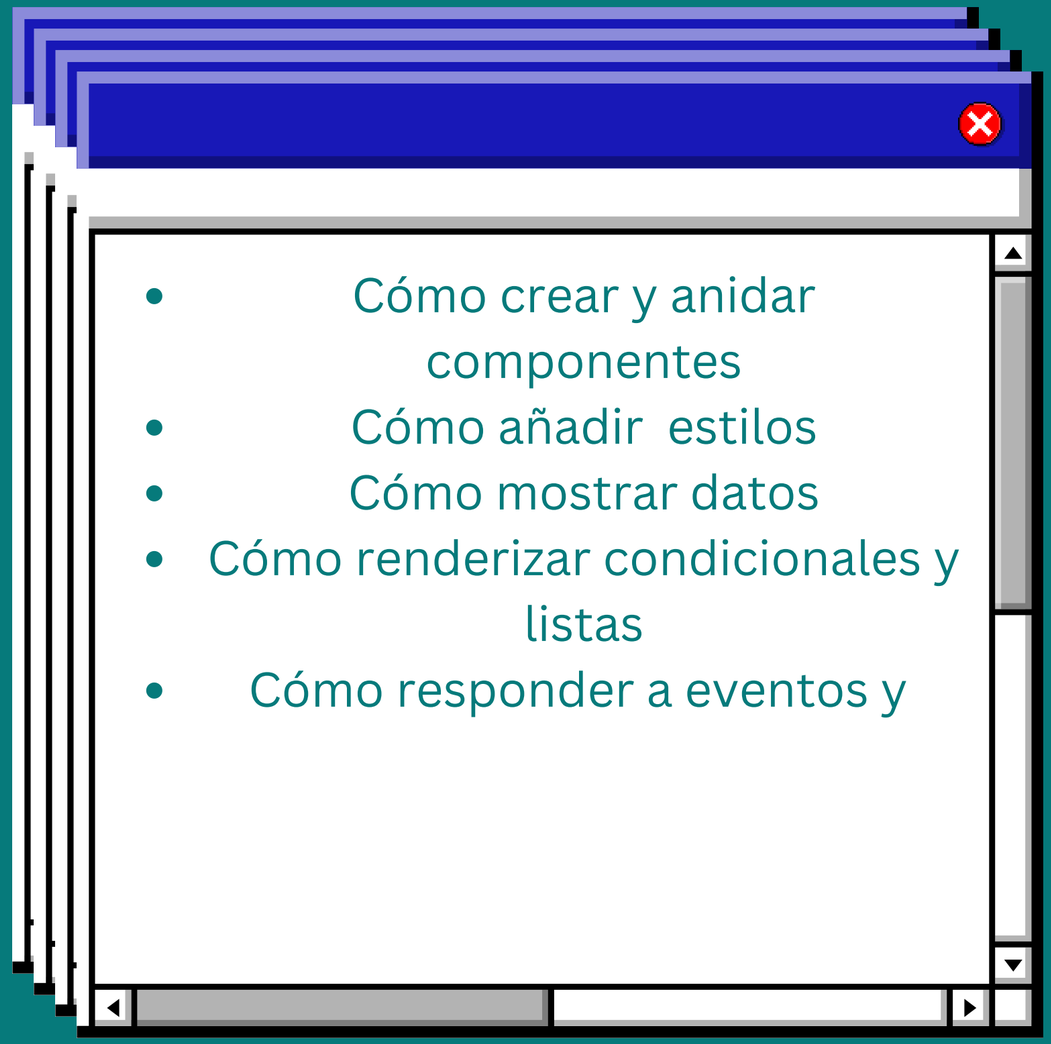


Instalar “create-react-app”

npm install -g create-react-app

```
npm start
```

Puntos importantes a abordar



- Cómo crear y anidar componentes
- Cómo añadir estilos
- Cómo mostrar datos
- Cómo renderizar condicionales y listas
- Cómo responder a eventos y

Back to Agenda Page

Crear y Añadir Componentes

Un componente es una pieza de UI (interfaz de usuario)

Un componente puede ser tan pequeño como un botón, o tan grande como toda una página.

Los componentes de React son funciones de JavaScript que devuelven markup

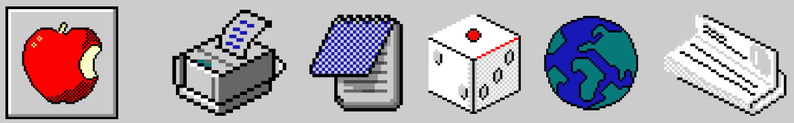
Ejemplo

```
function MyButton() {
  return (
    <button>Soy un botón</button>
  );
}
```

```
export default function MyApp() {
  return (
    <div>
      <h1>Bienvenido a mi aplicación</h1>
      <MyButton />
    </div>
  );
}
```

Bienvenido a mi aplicación

Soy un botón



Back to Agenda Page

Cómo añadir estilos

En React, especificas una clase de CSS con className.

```
<img className="avatar" />
```

Luego escribes las reglas CSS para esa clase en un archivo CSS aparte:

```
/* In your CSS */
.avatar {
  border-radius: 50%;
}
```

Mostrar Datos

Las llaves te permiten «escapar de nuevo» hacia JavaScript de forma tal que puedas incrustar una variable

También puedes «escaparte hacia JavaScript» en los atributos JSX, pero tienes que utilizar llaves en lugar de comillas.

Ejemplo

```
return (
  <h1>
    {user.name}
  </h1>
);
```

```
return (
  <img
    className="avatar"
    src={user.imageUrl}
  />
);
```

Hedy Lamarr

Back to Agenda Page

Renderizado condicional

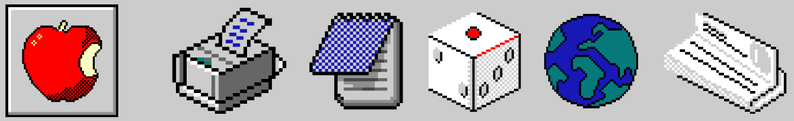
En React, no hay una sintaxis especial para escribir condicionales, Por lo cual se usan las mismas técnicas que usas al escribir código regular de JavaScript.

```
let content;
if (isLoggedIn) {
  content = <AdminPanel />;
} else {
  content = <LoginForm />;
}
return (
  <div>
    {content}
  </div>
);
```

Otra Forma

Una Forma mas compacta es utilizar el operador ? condicional. A diferencia de if, funciona dentro de JSX:

```
<div>
  {isLoggedIn ? (
    <AdminPanel />
  ) : (
    <LoginForm />
  )}
</div>
```



Back to Agenda Page

Renderizado de listas

Para esto dependerás de funcionalidades de JavaScript como los bucles for y la función map() de los arreglos para renderizar listas de componentes.

Ejemplo

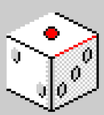
Arreglo de productos:

```
const products = [
  { title: 'Col', id: 1 },
  { title: 'Ajo', id: 2 },
  { title: 'Manzana', id: 3 },
];
```

Utiliza la función map() para transformar el arreglo de productos en un arreglo de elementos <li>:

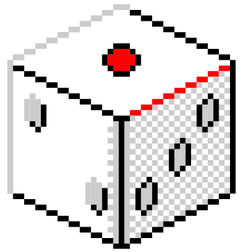
```
const listItems = products.map(product =>
  <li key={product.id}>
    {product.title}
  </li>
);

return (
  <ul>{listItems}</ul>
);
```



Back to Agenda Page

Responder a eventos



Puedes responder a eventos declarando funciones controladoras de eventos dentro de tus componentes

```
function MyButton() {
  function handleClick() {
    alert('¡Me hiciste clic!');
  }

  return (
    <button onClick={handleClick}>
      Hazme clic
    </button>
  );
}
```